

# ReadSpeaker

---

## ReadSpeaker VTAPI Android Guide

---

The software described in this manual is furnished under a valid license agreement with ReadSpeaker. The software may be used only in accordance with the terms of the agreement.

### Copyright Notice

Copyright © 2000-2019 ReadSpeaker All Rights Reserved. All documents issued by ReadSpeaker is the intellectual property of ReadSpeaker.

**Disclaimer.** THIS MANUAL IS DISTRIBUTED AS-IS. ReadSpeaker SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN; NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL. THE INFORMATION IN THIS MANUAL IS SUBJECT TO CHANGE WITHOUT ANY PRIOR NOTICE.

Copying or distributing whole or part of this document is not permitted without prior written permission of ReadSpeaker.

2019-02 Printed in Seoul, Republic of Korea

---

A open-source software was used to develop vtconv API.(APR-iconv library)

Copyright © 2000-2015 ReadSpeaker  
All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

OPUS Codec is used for P-type engine.

Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic,  
Jean-Marc Valin, Timothy B. Terriberry,

CSIRO, Gregory Maxwell, Mark Borgerding,  
Erik de Castro Lopo

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are specified at:

Xiph.Org Foundation:  
<https://datatracker.ietf.org/ipr/1524/>

Microsoft Corporation:  
<https://datatracker.ietf.org/ipr/1914/>

Broadcom Corporation:  
<https://datatracker.ietf.org/ipr/1526/>

---

## Table of contents

---

1. [Overview](#)
2. [Installation](#)
3. [Getting Started](#)
4. [Project Settings](#)
5. [License Key](#)
6. [Appendix A: Audio format available](#)

## 1. Overview

---

**ReadSpeaker** text-to-speech product by **ReadSpeaker** is the leading text-to-speech solution, which synthesizes an arbitrary input text to produce synthetic speech that is natural, human-like and clear by accurately analyzing and processing linguistic information such as phonetic content, sentence structure, prosody and pauses

As the proud leader of high quality, high performance voice engines in Korean, English, Chinese , Japanese and EU languages that are able to synthesize variety of text and symbols in best quality voice, ReadSpeaker is widely used in industries including telecommunications, finance service and public welfare. With the precise pronunciation, natural human-like synthesized voice, variety of supported platforms, high quality service, stability, fast synthesis speed and variety of voice types and language, ReadSpeaker proudly guarantees that ReadSpeaker is the best speech synthesis product in the world.

This document contains explanations and examples of VTAPI (VT Application Program Interface) of **ReadSpeaker** necessary for developing programs using multiple **ReadSpeaker** Engines .

## 2. Installation

---

### VTAPI package

**VTAPI** consists of jar file for API and native library. Copy it to project to use it.

#### jniLibs

- VTAPI Native Library
  - architectures : `armeabi-v7a`, `arm64-v8a`, `x86`, `x86_64`
  - optional library : `librstmwrapper.so`, `libvtconv.so`, `libvtsave.so`, `libvteffect.so`, `libcp[codepage].so`, `libvtssml.so`

#### libs

- `vtapi4a.jar` : In order to make it usable in Android SDK, it calls native library to use VTAPI.
- `vtlicensemodule.jar` : Used when a license key is available and used to download a license.

Caution: `libvteffect` library is not used in Android.

---

### Engine package

**Engine** consists of DB file, engine library, `vtpath.ini` or `rspath.ini`.

#### VT engine package file

- `bin(libraries)` [`language` : `eng`, `kor`, `jpn...`] [`type` : `d16`, `p16...`]

- architectures : consists of `armeabi-v7a`, `arm64-v8a`, `x86`, `x86_64`

Note: You as a developer must copy the **Engine library** files from these folders to the project `[jniLibs/[ARCH]/]`.

- `libvt_[language]_[type].so`
- **tts\_single\_db.vtdb2 : Engine DB file**
- **vtpath.ini : copy all texts and then add to one ini file if you have more than one engine.**
  - packname : consists of `[VoiceName]_[type]_setup_engine`
  - Path[VoiceID] : `db(.vtdb2)` path, default one is `Environment.getExternalStorageDirectory().getAbsolutePath()`. **[What user has to change]**
    - ex) Copy **[James]** folder which includes `tts_single_db.vtdb2` to a `Environment.getExternalStorageDirectory().getAbsolutePath()` when **[James]** folder is registered. Write `"Path104:/sdcard/Speech/D16/James"`
  - Lang : voice Language
  - CodePage : voice code page
  - Gender : voice gender
  - ISOCode : voice ISO Code (ex. en-US)
  - SamplingRate : voice sample rate

## RS engine package file

- **bin(libraries)**
  - architectures : consists of `armeabi-v7a`, `arm64-v8a`, `x86`, `x86_64`

Note: You as a developer must copy the **Engine library** files from these folders to the project `[jniLibs/[ARCH]/]`.

  - `librstts-2.so`
- **data(db) [ISOCode : en-US, en-GB...]**
  - `[VoideName].db`
  - `[VoideName].fon`
  - `[VoideName].opu`
  - `[VoideName].udb`
  - `[ISOCode].db`
  - `[ISOCode].fsa`
  - `[ISOCode].fst`
- **rspath.ini : copy all texts to one ini file if you have multiple engines.**
  - packname : consists of `[VoiceName]_[type]_setup_engine`
  - Path[VoiceID] : data folder path. Default path is a `Environment.getExternalStorageDirectory().getAbsolutePath()`. **[What user has to change]**

The path is considered as the path for data folder. It means that the 'data' folder should be under the engine name path. (for example, the data folder is under 'Alice' folder.)

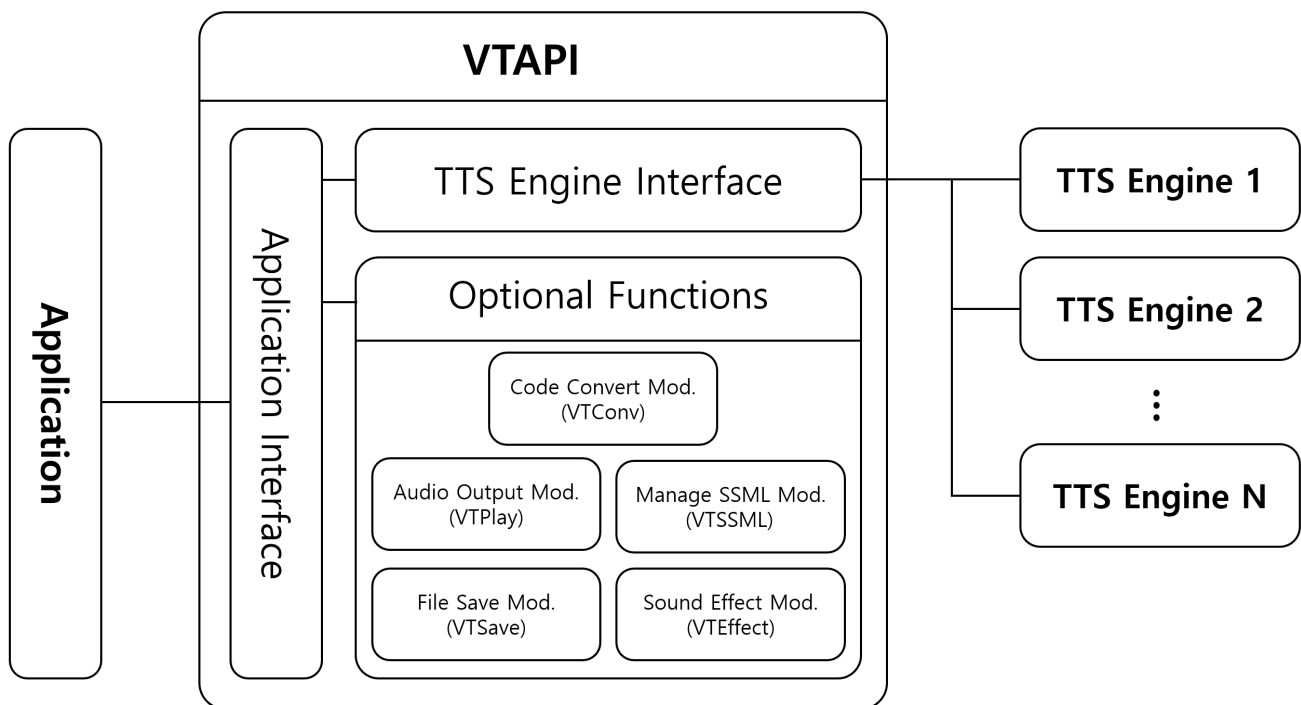
\* ex) Copy **[\*\*Alice\*\*]** folder which includes **[data]** folder to a publicly accessible path, `Environment.getExternalStorageDirectory().getAbsolutePath()` for example. When **[\*\*Alice\*\*]** folder is copied to the device, write its path in `rspath.ini`, `"Path5000:/sdcard/**Alice**/"`.

- Lang : voice Language
- CodePage : voice code page
- Gender : voice gender
- ISOCode : voice ISO Code (ex. en-US)
- SamplingRate : voice sample rate

### 3. Getting started

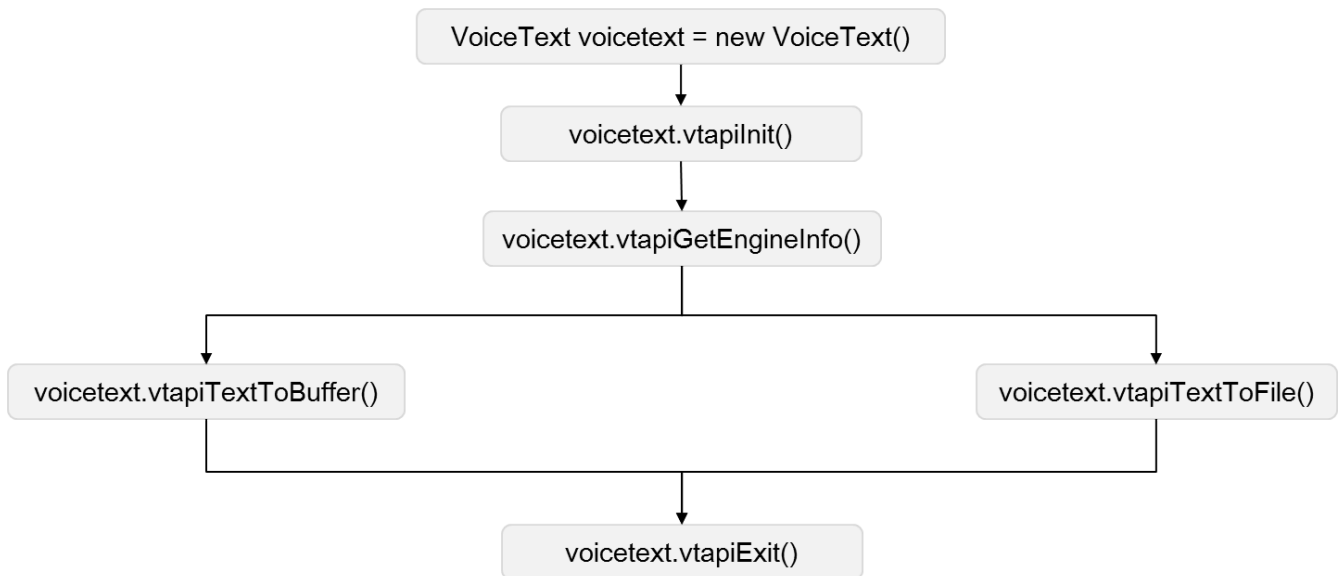
**VTAPI** is **High-Level API** of ReadSpeaker engine. The existing ReadSpeaker Engine API is provided differently for each language. VTAPI provides **integrated** API, making it possible to build an application with engines available but without a need to change an API in application.

VTAPI is structured as below.



### Work flows

A flow for Android VTAPI API



## VTAPI sample code

A flow for Android VTAPI API

```

public static final String rootPath =
Environment.getExternalStorageDirectory().getAbsolutePath() + "/HoyaSpeech/";
VoiceText voicetext = new VoiceText();
try {
    voicetext.vtapiInit(rootPath);
    Options attr = new Options();
    HashMap<String, EngineInfo> engines = voicetext.vtapiGetEngineInfo();
    EngineInfo firstEngine = null;
    for (Map.Entry<String, EngineInfo> e : engines.entrySet()) {
        System.out.print(e.getKey() + " - ");
        System.out.println(e.getValue().getSpeaker());
        firstEngine = e.getValue();
        break;
    }
    if (engines.isEmpty()) {
        return;
    }
    String testText = "Hillary Clinton has jumped into a 12-point lead against rival
Donald Trump in a new poll, pushed largely by disapproval of the Republican nominee. The
poll, one of many with a large deficit for Trump, shows his support dropping to a new low
as Clinton's soars";
    voicetext.vtapiTextToBuffer(testText, false, false, 320, firstEngine.getSpeaker(),
                                firstEngine.getSampling(),
                                firstEngine.getType(), attr,
                                Constants.OutputFormat.FORMAT_16PCM,
    new VoiceTextListener() {
        @Override
        public void onReadBuffer(byte[] output, int frameSize) {
            // Audio Play
        }
    }

    @Override

```

```

        public void onError(String reason) {
            //error
        }
    });
} catch (Exception e) {
    //error
}
voicetext.vtapiExit();

```

## [Android] VTAPI Application Programming Interface

Please see the attached java\_doc folder.

### 4. Project settings

VTAPI and **Engine** library should be registered to ▼ **Embedded Binaries**.

Add library (build.gradle)

```

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation files('libs/vtapi4a.jar')
    implementation files('libs/vtlicensemodule.jar') // Add only when a license key is
used
}

```

AndroidManifest.xml [uses-permission]

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="xxx.xxx.xxx">
    <application
        ...
    </application>
    <!-- VTAPI 및 vtlicensemodule.jar(LicenseKey) Permission for use -->
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <!-- vtlicensemodule.jar(LicenseKey) Permission for use -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
</manifest>

```

## Sample Code - Check For Permissions

```
// https://developer.android.com/training/permissions/requesting
@TargetApi(Build.VERSION_CODES.M)
public void checkForPermissions()
{
    /**
     * EXTERNAL STORAGE
     */
    if(ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED
        || ActivityCompat.checkSelfPermission(this,

Manifest.permission.READ_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED) {
        // Should we show an explanation?
        if(ActivityCompat.shouldShowRequestPermissionRationale(this,
            Manifest.permission.WRITE_EXTERNAL_STORAGE))
        {
            // Show an explanation to the user *asynchronously* -- don't block
            // this thread waiting for the user's response! After the user
            // sees the explanation, try again to request the permission.
        } else {
            // No explanation needed, we can request the permission.
            ActivityCompat.requestPermissions(this ,
                new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE,
                    Manifest.permission.READ_EXTERNAL_STORAGE}, 1);
        }
    }

    /**
     * ACCESS, CHANGE WIFI STATE
     */
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED
        || checkSelfPermission(Manifest.permission.ACCESS_COARSE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
            Manifest.permission.ACCESS_FINE_LOCATION)) {
            // ...
        } else {
            ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.ACCESS_COARSE_LOCATION}, 1);
        }
    }
}
```



## 5. License key

---

Used only when a license key is provided (LICENSE\_KEY : XXXX-XXXX-XXXX-XXXX-XXXX) Use vtlicensemodule.jar to create VtLicenseSetting and then do coding for downloading before calling **VoiceText.load()**

### Sample Code - License Download

```
String LICENSE_KEY = "xxxx-xxxx-xxxx-xxxx-xxxx";
String LICENSE_PATH = Environment.getExternalStorageDirectory().getAbsolutePath()
    + "/HoyaSpeech/db/verify/";

VtLicenseSetting licensemodule = new VtLicenseSetting(getApplicationContext());
//licensemodule.resetLicenseFileDownload();//retry download license
licensemodule.vtLicenseDownload(LICENSE_KEY, LICENSE_PATH,
    new LicenseDownloadListener() {
        @Override
        public void onSuccess() {
            Log.e("LICENSE", "onSuccess" );
        }

        @Override
        public void onFailure(String s) {
            Log.e("LICENSE", "onFailure : " + s );
        }

        @Override
        public void onError(String s) {
            Log.e("LICENSE", "onError : " + s );
        }
    });

VoiceText voicetext = new VoiceText();
voicetext.vtapiSetLicenseFolder(LICENSE_PATH);
voicetext.vtapiUsingLicensekey(true);
```

## 6. Appendix A : Audio format available

---

### Audio format available

Format	TTS Engine	TTS Server - VTAPI	TTS Server - File	TTS Server - One Frame Buffer	TTS Server - Multiple Frame Buffer
16bit linear PCM	O	O	O	O	O
8bit A-law PCM	O	O	O	O	O
8bit Mu-law PCM	O	O	O	O	O
4bit Dialogic ADPCM	O	X	O	O	O
16bit linear PCM Wave	O	O	O	O	O
8bit unsigned linear PCM Wave	O	O	O	O	X
8bit A-law PCM Wave	O	O	O	O	X
8bit Mu-law PCM Wave	O	O	O	O	X
ASF	X	X	O	O	X
OGG	X	X	O	O	X

## 7. Appendix B : List on errors

---

Name	Value	Descriptions
VTAPI_INVALID_PARAM_ERROR	-1	A parameter is incorrect.
VTAPI_INVALID_VALUE_ERROR	-2	A value for an internal structure variable is not valid.
VTAPI_INCORRECT_SET_ERROR	-3	An output type which is currently set does not match an API.
VTAPI_INVALID_ID_OR_THREAD_ERROR	-4	Either the engine handle or thread handle is not valid.
VTAPI_USERDICT_ERROR	-5	User dictionary error
VTAPI_SSML_SYNTAX_ERROR	-6	SSML syntax error
VTAPI_MEMORY_ALLOC_ERROR	-7	Memory allocation fails.
VTAPI_NOT_OPENBUFFER_ERROR	-8	The initialization is in trouble when buffer opens.
VTAPI_VTSSML_INVALID_ERROR	-9	No default voice exists as a result of parsing by VTParser.
VTAPI_INVALID_ENGINE_ERROR	-10	A selected engine is not found.
VTAPI_OVER_CHANNEL_ERROR	-11	The number of channels in a verified license is exceeded when requesting for synthesizing.
VTAPI_NODATA_ERROR	-12	No requested information exists.
VTAPI_NOT_SUPPORTED_ERROR	-13	No function available
VTAPI_DLL_VTPLAY_NOT_LINKING	-21	VTPlay module unfounded
VTAPI_DLL_VTSSML_NOT_LINKING	-22	SSML modue unfounded
VTAPI_DLL_VTSAVE_NOT_LINKING	-23	VTSave module unfounded.
VTAPI_DLL_VTCONV_NOT_LINKING	-24	VTConv module unfounded.
VTAPI_DLL_VTEFFECT_NOT_LINKING	-25	VTEffect module unfounded.
VTAPI_CHECKLICENSE_ERROR_NOTEXIST	-51	Either license file or buffer does not exist.
VTAPI_CHECKLICENSE_ERROR_INVALIDFORMAT	-52	Either license file or buffer format is not right.
VTAPI_CHECKLICENSE_ERROR_EXPIREDATE	-53	License date is expired.
VTAPI_CHECKLICENSE_ERROR_HOSTID	-54	Unable to get license HOSTID
VTAPI_CHECKLICENSE_ERROR_OS	-55	Unable to get license OS
VTAPI_CHECKLICENSE_ERROR_LANG	-56	Unable to get license LANG value
VTAPI_CHECKLICENSE_ERROR_SPEAKER	-57	Unable to get license voice name

Name	Value	Descriptions
VTAPI_CHECKLICENSE_ERROR_VERSION	-58	Unable to get license version
VTAPI_CHECKLICENSE_ERROR_DBACCESS	-59	Unable to get license DB access information
VTAPI_CHECKLICENSE_ERROR_SAMPLING	-60	Unable to get license sampling information
VTAPI_CHECKLICENSE_ERROR_DBSIZE	-61	Unable to get license DB size
VTAPI_LICENSEINFO_ERROR_NOT_SUPPORTED_REQUEST	-81	A unsupported request comes in getting license information
VTAPI_LICENSEINFO_ERROR_NOTEXIST_LICENSE	-82	No license file exists.
VTAPI_LICENSEINFO_ERROR_INVALID_LICENSE	-83	License file is not right one.
VTAPI_LICENSEINFO_ERROR_NULL_VALUE	-84	A requested value for license information is null.
VTAPI_LICENSEINFO_ERROR_SHORT_LENGTH_VALUE	-85	A value for requested license information is too short to keep the information.
VTAPI_LICENSEINFO_ERROR_UNKNOWN	-86	A unknown error happens so that it is not possible to get license information.
VTAPI_LICENSEINFO_CHECKLICENSE_FAIL	-87	License file is not verified yet.
VTAPI_GETLICENSECOMMENT_ERROR_NOTVERIFY	-91	A verification problem occurs in getting license comment information.
VTAPI_GETLICENSECOMMENT_ERROR_INVALIDCOMMENT	-92	License comment information is found inappropriate.
VTAPI_GETLICENSECOMMENT_ERROR_NULLBUFFER	-93	A space to save license comment information is null.
VTAPI_GETLICENSECOMMENT_ERROR_SHORTBUFFER	-94	A space to save license comment information is small.
VTENGINE_VERSION_RETURN_ERROR	-100	Failed to get an engine version
VTENGINE_INIT_ERROR	-111	Engine initialization fails.
VTENGINE_INVALID_PARAM_ERROR	-112	A parameter in an engine is incorrect.
VTENGINE_ID_NOTFOUND_ERROR	-113	Unable to send engine information by an engine ID
VTENGINE_INVALID_VALUE_ERROR	-114	A value in engine structure is not valid.
VTENGINE_GET_INFO_ERROR	-115	Unable to get information about an engine of choice
VTENGINE_OVER_CHANNEL_ERROR	-116	The number of verification channels is exceeded.

Name	Value	Descriptions
VTENGINE_LOAD_ALREADY_ERROR	-117	Engine already loaded
VTENGINE_OPEN_CONFIG_ERROR	-118	A problem happens when Config file being loaded.
VTENGINE_OPEN_REGISTRY_ERROR	-119	It is failed to get information about engine installation from registry from Windows.
VTENGINE_MALLOC_ERROR	-120	Memory fails to be allocated while an engine functions to synthesize.
VTENGINE_NOTLINK_API_ERROR	-121	Unable to find an API address in engine. (In case of an engine API newly added)
VT_LOADTTS_ERROR_CONFLICT_DBPATH	-131	Tried to do DB loading with the same voice through another db_path in using multiple DB
VT_LOADTTS_ERROR_TTS_STRUCTURE	-132	Failed to secure channel memory
VT_LOADTTS_ERROR_TAGGER	-133	Failed to do DB loading about morpheme analysis
VT_LOADTTS_ERROR_BREAK_INDEX	-134	Failed to do DB loading about break index
VT_LOADTTS_ERROR_TPP_DICT	-135	Failed to do DB loading about text preprocessing
VT_LOADTTS_ERROR_TABLE	-136	Failed to do DB loading about acoustic model
VT_LOADTTS_ERROR_UNIT_INDEX	-137	Failed to do DB loading about unit selection
VT_LOADTTS_ERROR_PROSODY_DB	-138	Failed to do DB loading about prosody model
VT_LOADTTS_ERROR_PCM_DB	-139	Failed to do DB loading about voice
VT_LOADTTS_ERROR_PM_DB	-140	Failed to do DB loading about pitch position information
VT_LOADTTS_ERROR_UNKNOWN	-141	For other reasons
VT_LOAD_USERDICT_ERROR_INVALID_INDEX	-151	dictidx is not a value in valid range.
VT_UNLOAD_USERDICT_ERROR_NULL_INDEX	-156	A dictionary pointed by dictidx is already unloaded.
VT_LOAD_USERDICT_ERROR_INDEX_BUSY	-152	dictidx of a dictionary is already loaded.
VT_UNLOAD_USERDICT_ERROR_INVALID_INDEX	-157	dictidx is not a value in valid range.

Name	Value	Descriptions
VT_LOAD_USERDICT_ERROR_LOAD_FAIL	-153	A dictionary fails to be loaded for either no existing dictionary file or no valid entry.
VT_UNLOAD_USERDICT_ERROR_UNKNOWN	-158	User dictionary unloading error - other unknown errors
VT_LOAD_USERDICT_ERROR_UNKNOWN	-154	User dictionary loading error - other unknown errors
VTAPI_INFO_ERROR_NOT_SUPPORTED_REQUEST	-191	Unsupported request
VTAPI_INFO_ERROR_INVALID_REQUEST	-192	A request value is not valid.
VTAPI_INFO_ERROR_NULL_VALUE	-193	A parameter to receive an information value is null.
VTAPI_INFO_ERROR_SHORT_LENGTH_VALUE	-194	A parameter to receive an information value is short in length.
VTAPI_INFO_ERROR_UNKNOWN	-195	A unknown error occurs when getting an information value.
VTPLAY_INIT_ERROR	-201	VTPlay Module initialization fails.
VTPLAY_CLOSE_ERROR	-202	VTPlay Module release fails.
VTPLAY_PLAY_ERROR	-203	A problem occurs when VTPlay module generates.
VTPLAY_STOP_ERROR	-204	A problem occurs when VTPlay module stops generating.
VTPLAY_PAUSE_ERROR	-205	A problem occurs when VTPlay module generation pauses.
VTPLAY_RESTART_ERROR	-206	A problem occurs when VTPlay module resumes generating.
VTPLAY_AUDIO_OPEN_ERROR	-211	Audio device open error
VTPLAY_AUDIO_CLOSE_ERROR	-212	Audio device close error
VTPLAY_AUDIO_RESTART_ERROR	-213	Audio device restart error
VTPLAY_AUDIO_PAUSE_ERROR	-214	Audio device pause error
VTSAVE_PARAM_ERROR	-301	A parameter in VTSave module is incorrect.
VTSAVE_VEAPI_ERROR	-302	A problem with modulation of VTSave module.
VTCONV_THREAD_INIT_ERROR	-401	The initialization of text conversion thread fails.

Name	Value	Descriptions
VTCONV_CP_LINKING_ERROR	-402	Code page conversion module is not initialized.
VTCONV_CP_INIT_ERROR	-403	Code page conversion module is failed.